

CLAIMS

1. A testing platform within a computing resource environment, the testing platform comprising:

 a test execution engine that receives input commands and initiates processing of the input commands;

 a test routine; and

 components that serve to adapt hardware and software interfaces of the computing resource environment to the test execution engine and test routine and that shield the test execution engine and test routine from dependencies on the hardware and software interfaces of the computing resource environment.

2. The testing platform of claim 1 wherein the hardware and software interfaces of the computing resource environment include interfaces to external hardware components and peripherals, external data storage and data I/O devices, communications hardware, operating system interfaces, and software interfaces to programs and routines.

3. The testing platform of claim 2 wherein the components that serve to adapt hardware and software interfaces of the computing resource environment to the test execution engine include:

 user I/O components that adapt the test execution engine to user I/O interfaces;

 result handling components that adapt the test execution engine to data output and presentation interfaces, including interfaces to disk files, printers, and databases; and

 computing resource components that adapt the test execution engine to operating system interfaces, including memory management interfaces and timer interfaces.

4. The testing platform of claim 2 wherein the testing platform can concurrently include and employ multiple user I/O components, multiple result handling components, and multiple computing resource components.

5. The testing platform of claim 2 wherein the components that serve to adapt hardware and software interfaces of a computing resource environment to the test routine include:

user I/O components that adapt the test routine to user I/O interfaces;

result handling components that adapt the test routine to data output and presentation interfaces, including interfaces to disk files, printers, and databases;

computing resource components that adapt the test routine to operating system interfaces, including memory management interfaces and timer interfaces; and

communications components that adapt the test routine to communications interfaces.

6. The testing platform of claim 5 wherein the testing platform can concurrently include and employ multiple user I/O components, multiple result handling components, multiple computing resource components, and multiple communications components.

7. The testing platform of claim 1 further including components that adapt the test execution engine to the test routine and that adapt the test routine to the test execution engine.

8. The testing platform of claim 7 wherein the components that adapt the test execution engine to the test routine include:

a test executor component that adapts the test execution engine to a test routine linked to test platform object code in a common executable;

a test executor component that adapts the test execution engine to a separate test routine executable that runs within the computing resource environment; and

a test executor component that adapts the test execution engine to an external test routine.

9. The testing platform of claim 7 wherein the components that adapt the test routine to the test execution engine include:

test link components that adapt the test routine to user I/O components, result handling components, computing resource components, and communications components.

10. The testing platform of claim 1 wherein multiple test routines can be concurrently handled by the test platform, and may be executed:

concurrently;
sequentially;
synchronously;
asynchronously; and
according to programmed execution patterns.

11. The testing platform of claim 10 wherein a test sequencing component handles launching and execution behavior of groups of test routines.

12. The testing platform of claim 1 wherein a mode component interprets user input as testing platform commands and dispatches appropriate routine calls to execute testing platform commands.

13. The testing platform of claim 12 wherein multiple mode components are included in the testing platform, with a single mode component active at each instant in time.

14. The testing platform of claim 13 wherein deactivation of a mode component and activation of another mode component may be elicited by an input command.

15. A method for flexibly, extensibly, and portably testing components, the method comprising:

providing a component to test;
developing a test routine that tests the component, shielding the test routine from dependencies on hardware and software interfaces by employing interfaces to adapter components within the test routine; and

running the test routine from a testing platform that includes a test execution engine shielded from dependencies on hardware and software interfaces by employing interfaces to adapter components within the test execution engine.

16. The method of claim 15 wherein employing interfaces to adapter components within the test routine further includes:

employing interfaces to user I/O components that adapt the test routine to user I/O interfaces;

employing interfaces to result handling components that adapt the test routine to data output and presentation interfaces, including interfaces to disk files, printers, and databases;

employing interfaces to computing resource components that adapt the test routine to operating system interfaces, including memory management interfaces and timer interfaces; and

employing interfaces to communications components that adapt the test routine to communications interfaces.

17. The method of claim 15 wherein employing interfaces to adapter components within the test execution engine further includes:

employing interfaces to user I/O components that adapt the test execution engine to user I/O interfaces;

employing interfaces to result handling components that adapt the test execution engine to data output and presentation interfaces, including interfaces to disk files, printers, and databases; and

employing interfaces to computing resource components that adapt the test execution engine to operating system interfaces, including memory management interfaces and timer interfaces.

18. The method of claim 15 further including:

employing interfaces to adapter components within the test execution engine that adapt the test execution engine to the test routine; and

employing interfaces to adapter components within the test routine that adapt the test routine to the test execution engine.

19. The method of claim 18 wherein employing interfaces to adapter components within the test execution engine that adapt the test execution engine to the test routine further includes:

employing interfaces to test executor components that adapt the test execution engine to a test routine linked to test platform object code in a common executable;

employing interfaces to test executor components that adapt the test execution engine to separate test routine executables; and

employing interfaces to test executor components that adapt the test execution engine to external test routines.

20. The method of claim 18 wherein employing interfaces to adapter components within the test routine that adapt the test routine to the test execution engine further includes:

employing interfaces to test link components that adapt the test routine to user I/O components, result handling components, computing resource components, and communications components.

21. The method of claim 15 further including employing within the test execution engine an interface to a test sequencing component that handles launching and execution orderings and synchronicities of groups of test routines.

22. The method of claim 15 further including employing within the test execution engine an interface to a mode component that interprets user input as testing platform commands and dispatches appropriate routine calls to execute testing platform commands.